

การเขียนโปรแกรม ด้วยภาษา C#

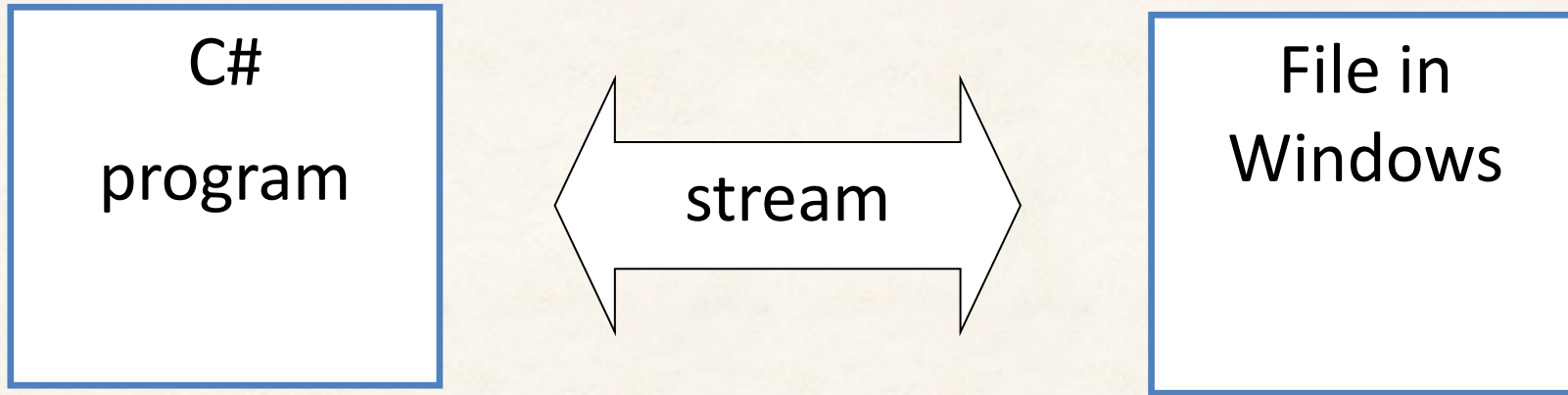
File Handling

File Handling

Files

- ข้อมูลต่างๆ จะหายไป เมื่อปิดโปรแกรม
- เราสามารถบันทึกลงแฟ้มข้อมูลก่อนที่จะปิดโปรแกรม เพื่อรักษาข้อมูลไว้บนสื่อเก็บข้อมูล
- ระบบปฏิบัติการ จะรับผิดชอบเรื่องการเขียนข้อมูลลงสื่อต่างๆ
- ในภาษา C# จะมี library รองรับเรื่องการทำงานกับแฟ้มข้อมูลไว้ให้แล้ว

Files and Streams



- ในการเขียนข้อมูลลงในไฟล์ เราใช้ stream
- stream จะช่วยให้เขียนข้อมูลลงในสื่อทุกชนิดในรูปแบบเดียวกันหมด (files, ports, network)

Creating a Stream

```
StreamWriter writer ;  
writer = new StreamWriter("hello.txt") ;
```

- บรรทัดแรก เป็นการสร้าง StreamWriter reference ชื่อ writer
 - StreamWriter มีความสามารถเขียนข้อมูลลงไฟล์
- บรรทัดที่สอง เป็นการสร้างวัตถุ stream แล้วอ้างถึงโดย writer
 - วัตถุ stream จะเชื่อมไปยังไฟล์ที่ชื่อ hello.txt
 - ไฟล์ชื่อ hello.txt ตั้งอยู่ที่ไหน?

Writing to a Stream

```
writer.WriteLine("hello world");
```

- StreamWriter มีเมธอดต่างๆ
 - จากตัวอย่าง จะเห็นว่ามี WriteLine เหมือนกับใน Console.WriteLine()
- โปรแกรมตัวอย่างนี้จะทำงานอย่างไร?

Closing a Stream

```
writer.Close();
```

- ทำไมต้องปิด stream ทุกครั้งที่เลิกใช้?
 - อาจเขียนข้อมูลลงไฟล์ไม่ครบ
 - อาจเรียกใช้ไฟล์จากโปรแกรมอื่นๆ ไม่ได้

คลาส Stream

- คลาส Stream เป็นส่วนหนึ่งของ C# Library
- อยู่ใน namespace ที่ชื่อ System.IO

```
using System.IO;
```

- หรือถ้าไม่ใช้ using เราต้องใช้แบบนี้

```
System.IO.StreamWriter writer;
```


File and Directories/Folders

- ระบบปฏิบัติการมีการเก็บระบบไฟล์เป็น directory
 - ไม่ได้เก็บทั้งหมดในที่เดียวกัน... ทำไม?
- เราเรียกว่า directory หรือ folder
 - ต่างกันตรงไหน?

Path

```
c:\data\2007\november\sales.txt
```

- Path คืออะไร?
- ในภาษา C# มีปัญหาเรื่องการเขียน \
 - เนื่องจาก C# มอง \ เป็น escape character
 - เราต้องใช้

```
string path;  
path = "c:\\data\\2007\\november\\sales.txt"
```

A Path as a string

```
string path;  
path = @"c:\data\2007\november\sales.txt";
```

ใน C# เราสามารถใช้ @ เพื่อไม่ต้องเขียน \\

File reading

```
TextReader reader =  
    new StreamReader("Test.txt");  
string line = reader.ReadLine();  
Console.WriteLine (line);  
reader.Close();
```

- อ่านไฟล์จาก Test.txt ที่ละบรรทัด
- แล้วพิมพ์ออกทาง console

Reading an Entire File

```
StreamReader reader;  
reader = new StreamReader("Test.txt");  
while (reader.EndOfStream == false)  
{  
    string line = reader.ReadLine();  
    Console.WriteLine(line);  
}  
reader.Close();
```

○ EndOfStream ใช้ตรวจสอบว่าจบไฟล์หรือยัง

สรุป

- เราใช้ stream เพื่อเชื่อมต่อระหว่างโปรแกรมและไฟล์บนสื่อเก็บข้อมูลต่างๆ
- stream ใช้ได้ทั้งเขียนและอ่าน
- การใช้งาน stream ถ้าไม่ระบุ path จะได้ไฟล์อยู่ที่เดียวกับโปรแกรมที่เราเขียน